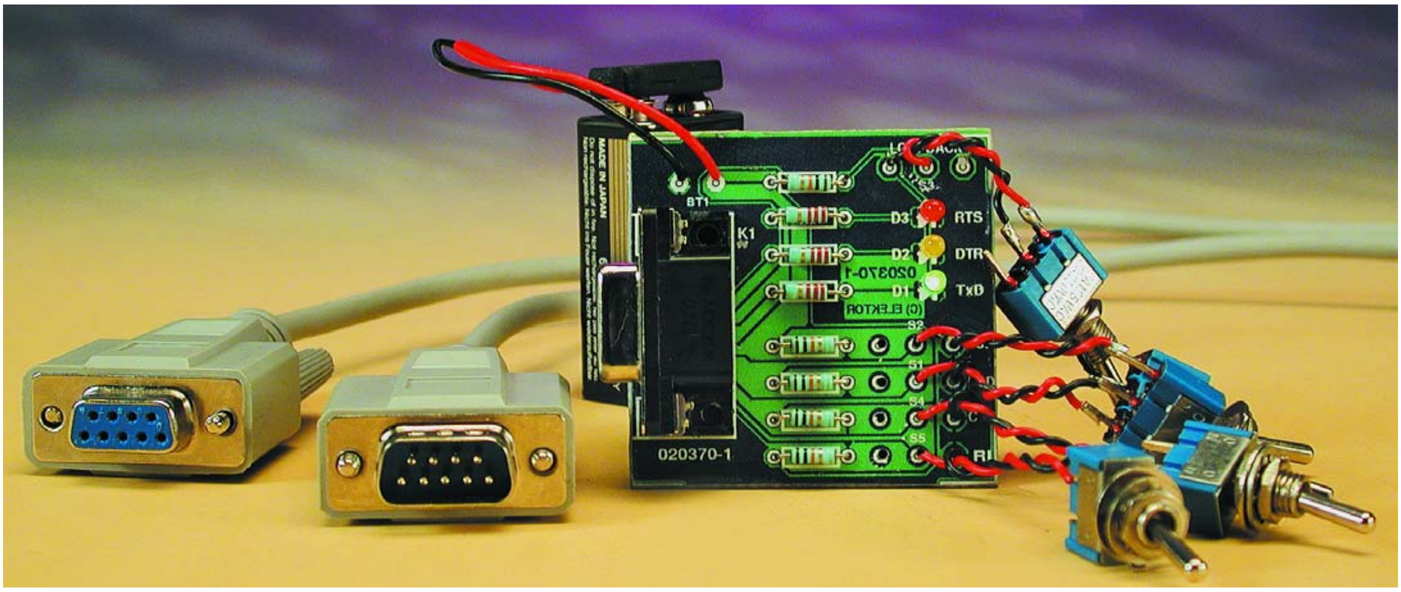
COM Port Tester

With 3 LEDs and 4 switches

Design by P. Goossens

The serial port (often called the COM port) is perfect for connecting various electronic circuits to the PC. The complexity of the connected circuit varies from a few I/O lines to control something and read the state of switches, to a complex communications system between several computers. This circuit is in the first category and uses the serial port to perform a few simple I/O tasks.



The COM Port Tester is a simple circuit that shows how the signals on a serial port can be used to control another electronic circuit. We will have to disappoint those of you who expect to use the serial port to control motors and lights etc. This circuit is limited to driving three LEDs and reading the state of four switches, nothing more and nothing less.

RS-232

The functions and electrical characteristics of the signals in a serial port have been laid down in a standard. This standard is called the RS-232 standard and was drawn up early in 1960 by the ‘Electronics Industries Association’. The aim was to define a standard port that could be used with equipment from various manufacturers to communicate between each other. A good 40 years later this standard (although with a few modifications) is still in widespread use in all kinds of equipment. The best known of these are of course the mouse, modem, etc., but in industrial equipment the presence of an RS-232 port is also common.

In the RS-232 standard equipment has been divided into two distinct groups, these being DTE and DCE. These abbreviations stand for ‘Data Terminal Equipment’ and ‘Data Communication Equipment’ respectively. These two terms were chosen by the authors of the standard due to the way in which the serial port was used at that time: on one side of the connection was a terminal, which was nothing more than a screen with a keyboard, and on the other end of the link was some sort of communication equipment.

The function of a terminal (DTE) is very simple in this set-up: every key that is pressed is transmitted via the serial link, and every character received is displayed on the screen. The communication equipment (DCE) takes care of all other aspects of the communication. The serial port is obviously not limited to the connection of terminals to communication equipment. Even so, DTE and DCE are still used to describe the two distinct groups of equipment.

Signals

The RS-232 standard describes many more signals than are required on the PC's serial port nowadays. This made it possible to use a 9-way connector instead of the original 25-way connector. The extra signals that were found on the 25-way connector are (almost) never used in modern equipment. Those signals that are in use can be found in **Table 1**. The direction of the signals is also included. 'DCE to DTE' means that the signal originates from the DCE (output) and ends up at the DTE (input). 'DTE to DCE' obviously means the opposite. The two most important signals are RxD and TxD. These signals are used for the exchange of the serial information ('serial' meaning one after the other, and in the case of the RS-232 port it is bit by bit).

A DTE can be easily distinguished from a DCE by looking at the connector used. If the equipment has a male connector for the serial port, then it is a DTE (for example a PC). If instead the equipment has a female connector, then the equipment concerned is a DCE (for example a modem, home-built circuit, etc.).

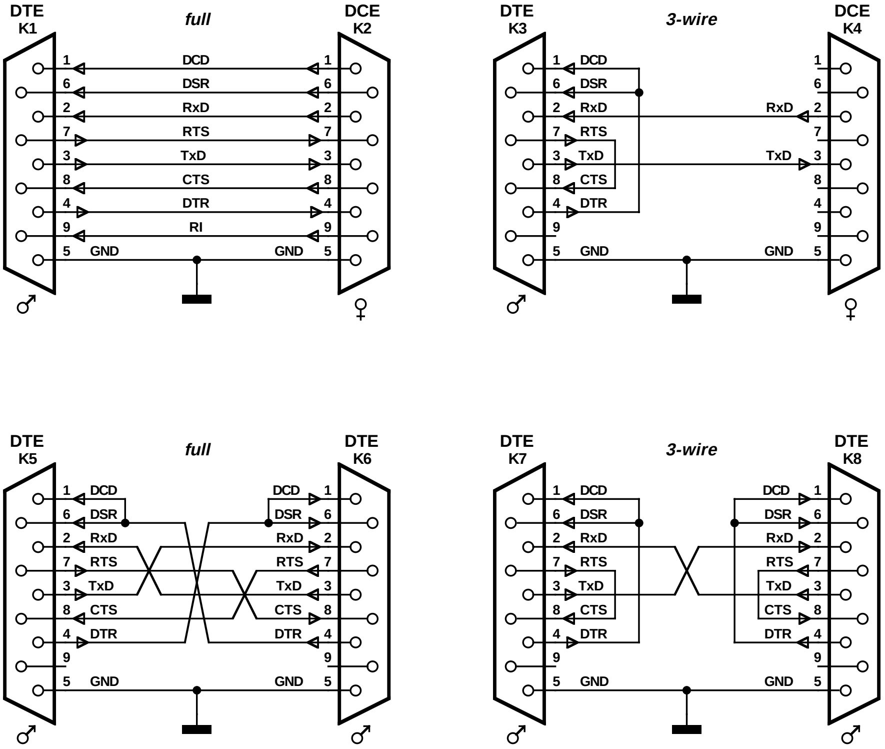


Figure 1. Several types of cable can be used to make a serial connection.

Electrical aspect

The signals on the RS-232 port obviously have to conform to an electrical specification. This is where the authors of the standard have decided on something that may appear peculiar to the digitally minded amongst you. A logical one (called a MARK in serial communications) is electrically represented by a signal between –3 V and –25 V, whereas a logical zero (RS-232 term: SPACE) has a voltage level between +3 V and +25 V. So not only are the voltages different compared to TTL levels, but all signals are also inverted.

Most PCs are now quite happy with TTL level signals, where a voltage between 0 V and 0.8 V corresponds to a '1', and a voltage between 2 V to 5 V corresponds to a '0'. Of course the modern RS-232 ports can still withstand voltages between –25 V and +25 V, so that equipment that conforms to the RS-232 specification won't cause any damage!

Cabling

Choosing the cable required to connect two pieces of equipment together via the serial port often causes confusion due to the many possible permutations. There is a big difference between connecting two DTEs (two PCs for example) and connecting a DCE to a DTE (a modem to a PC for example). Some of the different configurations can be seen in **Figure 1**.

When you examine these cables it is noticeable that for connections between a DTE and DCE no signals cross over, whereas for connections between two DTEs some signals do cross over. The number of signals that is required to make a connection between two pieces of equipment can vary from three to nine. The 25-way version has even more signals that can be connected. These signals have been ignored here for the sake of simplicity. They are also no longer very important, since 25-way connectors are hardly ever

Table 1. The signals on a 9-way connector.

DCD (Data Carrier Detect)	DCE to DTE
RxD (Receive Data)	DCE to DTE
TxD (Transmit Data)	DTE to DCE
DTR (Data Terminal Ready)	DTE to DCE
DSR (Data Set Ready)	DCE to DTE
RTS (Request To Send)	DTE to DCE
CTS (Clear To Send)	DCE to DTE
RI (Ring Indicator)	DCE to DTE

used any more for serial connections.

Usually a 3-wire connection is sufficient to achieve communications, but some equipment requires that all signals are connected. These extra signals are used for hardware handshaking, which we won't cover in this article.

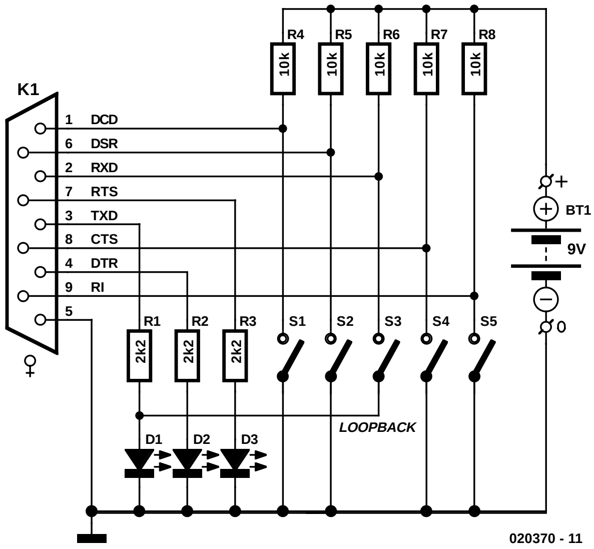


Figure 2. The circuit of the COM port tester consists of just a few resistors, switches and LEDs. A 9 V battery is used for power.

Circuit

The COM port tester (**Figure 2**) has been kept very simple. The main reason for this is that we did not keep to the exact RS-232 standard, which has a voltage between –3 V and –25 V for a logical one and a voltage between +3 V and +25 V for a logical zero. As was mentioned earlier, PCs are quite happy with TTL levels, so there is no longer a need to use a negative voltage to send a logical one to the PC.

All of the inputs and outputs found on a 9-way connector have been used in this circuit to communicate with the outside world. Here that was implemented using a few switches and LEDs.

An extra function is provided by switch S3. When this switch is closed, all serial data transmitted from the output of the serial port is fed back to the input of the serial port.

Figure 3 shows the layout of the PCB for this circuit. Considering the small number of components required, it shouldn't be a problem if you used a small prototyping board instead.

The circuit is connected to the PC via a one-to-one cable (the top-left cable shown in **Figure 1**).

Software

The original purpose of this circuit was to serve as an example to show how data could be processed in a simple manner via the serial port. Everybody will have their own ideas how this circuit can be put to use. This requires a program that has been adapted to the specific functions of the circuit.

The writing of such a program will be left as an exercise to the reader, but to start you off there is an article elsewhere in this issue of Elektor Electronics about a newly developed serial port driver for Windows. This is of course mainly of interest to people with some programming experience.

An example program for this article is obtainable from the publishers (order number **020388-11**, available on floppy disk or as a free download from the *Elektor Electronics* website). This permits the outputs of the serial port to be controlled manually, and also shows the state of the inputs.

After starting the program you can select the serial port required. Next, the port has to be opened by clicking on the button marked 'open'. The program is now ready for use and will show the state of the inputs using a tick in the boxes to the right of the connector.

When a box contains a tick it indicates that the relevant input is at a low voltage, which for an RS-232 signal means a logical one. This corresponds to a closed switch in our circuit.

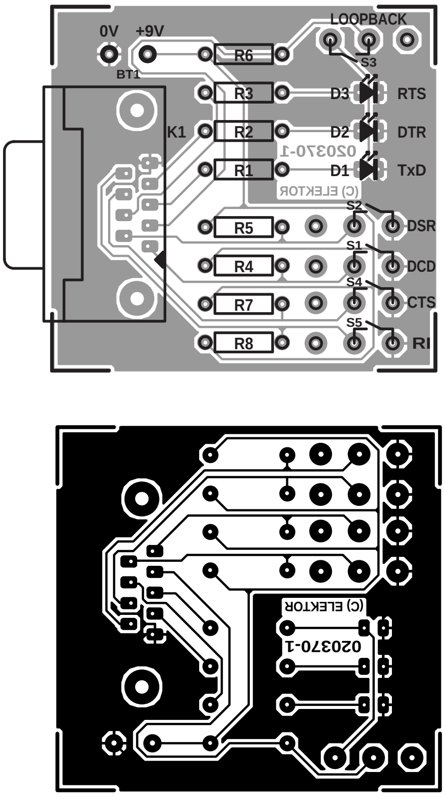


Figure 3. The PCB layout for the circuit in Figure 2 (board available from The PCBShop).

COMPONENTS LIST

Resistors:
R1,R2,R3 = 2kΩ2
R4-R8 = 10kΩ

Semiconductors:
D1 = LED, 3mm, green, low-current
D2 = LED, 3mm, yellow, low-current
D3 = LED, 3mm, red, low-current

Miscellaneous:
K1 = 9-way sub-D socket (female), angled pins, PCB mount
S1-S5 = slide switch or rocker switch, 1 make contact
9-V battery with clip-on leads
PCB, available from **The PCBShop**
Disk, project software, order code **020388-11** (see article 'Serial Port Driver for Windows')

Naming of RS-232 signals in electronic circuits

Often the naming of RS-232 signals in electronic circuit diagrams is misused. A female connector is frequently used for the serial port in a circuit, which means that the equipment should be considered as a DCE. This implies amongst other things that the pin where the serial data comes out should really be called RxD, but is usually identified in the circuit diagram as TxD. The opposite applies to the pin that receives serial data; this is usually called RxD, whereas the correct RS-232 term is TxD.

The circuit diagrams in Elector Electronics always show the pin that transmits serial data as TxD and the receiving pin as RxD, irrespective of the type of connector used.

This circuit is an exception to this 'rule', since there is no logical reason to exchange signals such as TxD and RxD in this case.

The three buttons to the right of the outputs of the connector are used to set the outputs high or low. This is made visible on our circuit via the LEDs.

Conclusion

This circuit has been kept very simple. Because of this, it only has a limited functionality. To increase its usefulness you could for example add a buffer stage to the outputs, which could then drive a relay for switching lamps. The switches could be replaced by an input stage with an opto-isolator, or a mercury switch etc., thereby allowing the acceptance of different information by the serial port of the PC.

These improvements have not been included here, since we wanted to keep the circuit small and simple. But there is nothing that stops you from extending the circuit yourself, for example turning it into a computer controlled time switch, burglar alarm, and so on. The possibilities are only limited by your imagination (and your knowledge of electronics of course!).

(020370-1)